$M^3Fusion$: Un modèle d'apprentissage profond pour la fusion de données satellitaires Multi-{Échelles/Modalités/Temporelles}

P. Benedetti[1] R. Gaetano[2,4] K. Ose[1] R. Pensa[5] S. Dupuy[3,4] D. Ienco[1]

[1] IRSTEA, UMR TETIS, Univ. Montpellier, Montpellier, France
[2] CIRAD, UMR TETIS, 500 Rue J.-F. Breton, F-34000 Montpellier, France
[3] CIRAD, UMR TETIS, F-97410 Saint-Pierre, Réunion, France
[4] UMR TETIS, Univ. Montpellier, AgroParisTech, CIRAD, CNRS, IRSTEA, Montpellier, France
[5] Dept. of Computer Science, University of Turin, Turin, Italy

{paola.benedetti, kenji.ose, dino.ienco}@irstea.fr
{raffaele.gaetano, stephane.dupuy}@cirad.fr
ruggero.pensa@unito.it

Résumé

Les systèmes modernes d'observation de la Terre fournissent des données à différentes résolutions temporelles et spatiales. Parmi les capteurs optiques, la constellation Sentinel-2 acquiert aujourd'hui des images à haute résolution temporelle (tous les 5 jours) et à haute résolution spatiale (10 m) qui sont utiles pour étudier la dynamique de l'occupation du sol. D'autre part, les images à très haute résolution spatiale (THRS) demeurent des données essentielles pour identifier les éléments caractérisés par des motifs spatiaux fins. Comprendre comment exploiter efficacement l' hétérogénéité des informations fournies par différents capteurs, dans un processus de fusion de données, est un défi majeur dans le domaine de la télédétection.
En utilisant conjointement les séries temporelles à haute résolution et les informations spatiales fines contenues dans les images THRS, nous proposons un modèle d'apprentissage profond, appelé $M^3Fusion$, pour relever ce défi et cartographier ainsi l'occupation du sol.
Les expériences, menées sur l'Île de la Réunion, montrent la qualité de notre architecture neuronale, comparée à une approche standard d'apprentissage automatique, tant en termes de performances quantitatives que de rendu spatial.

Mots Clef

Occupation du sol, Fusion de données, Apprentissage Profond, Série Temporelle d'images satellitaires, Très Haute Résolution Spatiale.

Abstract

Modern Earth Observation systems provide remote sensing data at different temporal and spatial resolutions. As regards optical sensors, nowadays, the Sentinel-2 program supplies images with high temporal resolution (every 5 days) and high spatial resolution (10m) that can be useful to monitor land cover dynamics. On the other hand, Very High Spatial Resolution images (VHSR) are still an essential data to figure out land cover mapping characterized by fine spatial patterns. Understand how to effectively leverage together these complementary sources of information to deal with land cover mapping is still challenging.
With the aim of dealing with land cover mapping through the fusion of multi-temporal High Spatial Resolution and Very High Spatial Resolution satellite images, we propose an End-to-End Deep Learning framework, named $M^3Fusion$, able to leverage simultaneously the temporal knowledge contained into time series data as well as the fine spatial information available in VHSR images.
Experiments carried out on the Reunion Island study area asses the quality of our proposal considering both quantitative and qualitative aspects. Finally, the performances of our method are also compared to a standard machine learning approach (Random Forest) used as baseline.

Keywords

Land Cover Mapping, Data Fusion, Deep Learning, Satellite Image Time series, Very High Spatial Resolution.

1 Introduction

Les programmes d'observation de la Terre produisent chaque jour d'énormes volumes de données. Ces informations peuvent être organisées en séries temporelles d'images satellitaires (SITS) à haute résolution (i.e. Sentinel) qui peuvent être utiles pour surveiller des territoires avec une dimension temporelle. En plus de cette information à haute fréquence temporelle, nous pouvons aussi obtenir de l'information à Très Haute Résolution Spatiale (THRS), comme par exemple de l'imagerie Spot6/7 ou

Pléiades, avec une fréquence temporelle plus limitée [15] (par exemple un fois par an).

L'analyse des séries temporelles et son couplage/fusion avec de l'information ponctuelle de type THRS restent un défi très important dans le domaine de la télédétection [12, 20].

Dans le contexte de la classification de l'occupation du sol, l'exploitation de séries temporelles d'images satellites à haute résolution spatiale (HRS), au lieu d'une seule image de la même résolution, est essentiel pour la distinction de certaines classes en fonction de leur profil temporel [1]. D'un autre côté, l'intégration d'une information spatiale plus fine permet de différencier certains aspects en tenant compte du contexte spatial à une échelle plus adéquate [20].

Généralement, les approches qui utilisent ces deux types d'information [10, 14], réalisent une fusion au niveau des descripteurs [20]. Ce type de fusion consiste à extraire un ensemble de descripteurs indépendants pour chaque source de données (série temporelle, image THRS). Ensuite ces descripteurs sont empilés ensemble et utilisés comme entrées pour une méthode classique d'apprentissage automatique supervisé (i.e., Random Forest).

Récemment, la révolution de l'apprentissage profond [21] a montré que les modèles de réseaux neuronaux sont des outils bien adaptés pour gérer et classifier automatiquement les données de télédétection [21]. La caractéristique principale de ce type de modèle est de pouvoir extraire, dans un seul processus, des descripteurs optimisés pour améliorer la classification et le classifieur associé. Cet avantage est fondamental dans un processus de fusion des données comme celui entre des séries temporelles à haute résolution spatiale (i.e. Sentinel-2) et des images à THRS (i.e. Spot6/7 et/ou Pléiades).

Parmi les méthodes d'apprentissage profond, nous pouvons distinguer deux grandes familles d'approches : les réseaux de neurones convolutifs [21] (CNN) et les réseaux de neurones récurrents [2] (RNN). Les réseaux CNN sont bien adaptés pour modéliser l'autocorrélation spatiale présente dans une image alors que les réseaux RNN sont spécialement conçus pour pouvoir gérer correctement des dépendances temporelles longues et complexes [9] issues d'une série temporelle multidimensionnelle.

Dans cet article, nous proposons de coupler les deux modèles (CNN et RNN) pour pouvoir aborder le problème de la fusion entre une série temporelle HRS et une image THRS sur la même zone d'étude afin de produire une classification de l'occupation du sol. Dans ce cadre, notre proposition vise à effectuer une fusion Multi-Échelles, Multi-Modalités et Multi-Temporelle. La méthode que nous proposons, nommée $M^3Fusion$ (Fusion Multi-Échelle/Modalités/Temporelles), prévoit une architecture de deep learning qui intègre à la fois une composante CNN (pour gérer l'information THRS) et une composante RNN (pour analyser l'information SITS à HRS) dans un unique modèle structuré d'apprentissage ("de bout en bout"). Chaque source d'information est intégrée à travers son module dédié et les descripteurs extraits sont ensuite concaténés pour effectuer la classification finale. Le fait de mettre en place un tel processus, qui prend en entrée les deux sources de données en même temps, nous garantit une extraction de descripteurs complémentaires et optimisés pour classifier in fine l'occupation du sol.

Pour valider notre approche, nous avons mené des expériences à partir d'un jeu de données sur l'Île de la Réunion, département français d'outre-mer localisé dans l'Océan Indien (à l'est de Madagascar), qui sera décrit dans la section 2. Le reste de l'article est organisé comme suit : la Section 3 présente l'architecture d'apprentissage profond ($M^3Fusion$) pour le processus de classification multi-source, le contexte expérimental et les résultats sont discutés dans la section 4 et des conclusions sont tirées dans la section 5.

2 Les Données

L'étude a été effectuée sur l'Île de la Réunion, département français d'outre-mer situé dans l'Océan Indien. Le jeu de données utilisé consiste en une série temporelle de 34 images Sentinel-2 (S2) acquises entre Avril 2016 et Mai 2017, ainsi qu'une image à très haute résolution spatiale (THRS) SPOT6/7 acquise en Avril 2016 et couvrant l'ensemble de l'île. Les images S2 utilisées sont celles fournies au niveau 2A par le pôle Surfaces Continentales THEIA [1], où les bandes à 20 m de résolution ont été rééchantillonnées à 10 m. Un traitement a été effectué pour combler les pixels masqués par des nuages via une interpolation linéaire multi-temporelle sur chaque bande (cfr. *Temporal Gapfilling*, [10]), et six indices radiométriques ont été calculés pour chaque date (NDVI, NDWI, indice de luminosité - BI, NDVI et NDWI de moyen infrarouge - MNDVI et MNDWI, et indice de végétation Red-Edge - RNDVI [10, 14]). Un total de 16 variables (10 réflectances de surface plus 6 indices) sont considérées pour chaque pixel de chaque image de la série temporelle.

L'image SPOT6/7, constituée à l'origine d'une bande panchromatique à 1.5 m et 4 bandes multispectrales (bleu, vert, rouge et proche infrarouge) à 6 m de résolution, a été fusionnée pour obtenir une seule image multispectrale à 1.5 m de résolution, puis rééchantillonnée à 2 m pour des exigences d'architecture du réseau d'apprentissage [2]. Sa taille finale est de 33280 $\times$ 29565 pixels sur 5 bandes (4 réflectances *Top of Atmosphere* plus le NDVI). Cette image a été également utilisée comme référence pour effectuer le recalage des différentes images de la série temporelle à l'aide d'une technique de recherche et de mise en correspondance de points d'amer et ce, afin d'améliorer la cohérence spatiale entre les différentes sources.

1. Données disponibles via `http://theia.cnes.fr`, prétraitées en réflectance de surface via le *MACCS-ATCOR Joint Algorithm* [5] développé par le Centre National d'Études Spatiales (CNES).

2. Ceci a été réalisé pour garantir une correspondance directe et sans chevauchements entre les pixels de série temporelle (10 m) et un bloc de pixels à THRS (5 $\times$ 5).

La base de données de terrain a été construite à partir de différentes sources : (i) la base de données du registre parcellaire graphique (RPG) de 2014, (ii) des relevés GPS réalisés en juin 2017 et (iii) une photo-interprétation pour les espaces naturels et les espaces urbains réalisée sur l'image THRS par un expert connaissant le territoire. Tous les contours des polygones ont été repris en utilisant comme réference l'image THRS. Le jeu de données de référence final comprend un total de 322 748 pixels (2 656 objets) répartis sur 13 classes, comme indiqué dans le Tableau 1.

Classe	Étiquette	**# Objets**	**# Pixels**
0	*Cultures maraichères*	380	12090
1	*Canne à sucre*	496	84136
2	*Vergers*	299	15477
3	*Plantations forestières*	67	9783
4	*Prairies*	257	50596
5	*Forêt*	292	55108
6	*Savane arbustive*	371	20287
7	*Savane herbacée*	78	5978
8	*Roches nues*	107	18659
9	*Zones urbanisées*	125	36178
10	*Cultures sous serre*	50	1877
11	*Surfaces en eau*	96	7349
12	*Ombres dues aux reliefs*	38	5230

TABLE 1 – Characteristics of the Reunion Dataset

3 Contributions

3.1 Description du modèle $M^3Fusion$

La figure 3 résume visuellement l'approche $M^3Fusion$ proposée dans cet article. Tout d'abord nous définissons quelles sont les données d'entrées de notre modèle d'apprentissage profond. Le réseau $M^3Fusion$ prend en entrée un jeux de données $\{(x_i, y_i)\}_{i=1}^{M}$ où chaque exemple est associé à une valeur de classe $y_i \in 1, ..., C$. Un exemple x_i est défini comme un couple $x_i = (ts_i, patch_i)$ tel que ts_i est la série temporelle (multidimensionnelle) d'un pixel Sentinel-2 (à 10m de résolution) et $patch_i$ est un morceau d'image Spot6/7 (à 2m de résolution) centrée autour du pixel Sentinel-2 correspondant. Chaque exemple a donc deux modalités de représentation : une modalité temporelle à haute résolution spatiale (fournie par la série temporelle d'images Sentinel-2) et une modalité à très haute résolution spatiale (fournie par l'image Spot6/7).

Pour chaque $patch_i$, nous avons fixé sa taille à une fenêtre de 25×25 pixels sur l'image Spot6/7 (qui correspond à des morceaux d'image Sentinel-2 de taille 5x5) centrés autour du pixel Sentinel-2 décrit par la série temporelle correspondante ts_i.

Pour pouvoir fusionner l'information temporelle de la série d'images Sentinel-2 et celle à très haute résolution spatiale contenue dans l'image Spot 6/7, nous avons conçu un réseau qui, en entrée, a deux branches complémentaires spécialisées pour chacune des deux modalités (spatiale/temporelle). Concernant la série temporelle du pixel Sentinel-2, nous gérons celle-ci à l'aide d'une architecture de Réseau de Neurones Récurrent (RNN). Plus particulièrement, nous avons utilisé une Gated Recurrent Unit (GRU) introduite en [4] et qui a déjà montré son efficacité dans le domaine de la télédétection [17, 16]. Par contre, l'information spatiale, à la résolution de 2m, introduite à travers l'imagerie à THRS est intégrée grâce à l'utilisation d'un Réseau de Neurones Convolutif [15] qui permet d'extraire les informations utiles par rapport au contexte spatial du pixel Sentinel-2 à classer.

Après cette première étape, les deux groupes de descripteurs entraînés à travers les deux flots d'analyse menés en parallèle sont utilisés pour établir la classification finale. Cette dernière est produite à la résolution du pixel Sentinel-2. En suivant la philosophie introduite en [7], l'architecture proposée vise en particulier à apprendre deux lots de descripteurs complémentaires (grâce aux différentes modalités spatiale et temporelle) qui soient le plus discriminant possible notamment lorsqu'ils sont utilisés indépendamment. Pour garantir ce dernier point, la stratégie prévoit l'ajout de deux classifieurs auxiliaires, travaillant chacun sur un groupe de descripteurs, comme montré dans le schéma de la Figure 3. Un troisième classifieur, travaillant sur la fusion (par concaténation) des deux groupes de descripteurs, fournit en sortie la classification finale. Chacun des classifieurs susmentionnés est réalisé en reliant directement ses descripteurs aux neurones de sortie à travers une fonction SoftMax [21] communément utilisée par la tâche de classification multi-classe dans l'apprentissage profond. L'apprentissage du modèle $M^3Fusion$ est réalisé en utilisant une combinaison linéaire de fonctions de coût de chaque classifieur pour la rétro-propagation de l'erreur.

3.2 Intégration de l'information de la série temporelles d'images à Haute Résolution Spatiale (STIS)

Récemment, des approches de réseaux des neurones récurrent (RNN) ont montré leurs qualités dans le domaine de la télédétection pour produire une occupation du sol à partir de séries temporelles d'images optiques [9] et reconnaître l'état du couvert végétal à l'aide de séries d'images satellitaires radar Sentinel-1 [16]. Motivés par ces récents travaux, nous avons décidé d'introduire un module RNN pour pouvoir intégrer l'information provenant de la série temporelle Sentinel-2 dans notre processus de fusion basé sur un apprentissage profond. Dans notre proposition, nous avons choisi le modèle GRU (Gated Recurrent Unit) introduit dans [4] que nous avons couplé à un mécanisme d'*attention* [3]. Les mécanismes d'attention sont très utilisés dans le traitement automatique du signal (langage ou signal 1D) et permettent de combiner entre eux les informations extraites, par le modèle GRU, aux différentes estampilles temporelles. L'entrée d'une unité GRU est une séquence (x_{t_1},..., x_{t_N}) où un élément générique x_{t_i} est un vecteur multidimensionnel de caractéristiques et t_i renvoie à la date correspondante de la série temporelle. Comme sortie du modèle GRU, nous avons une séquence de vecteurs de descripteurs entraînés à

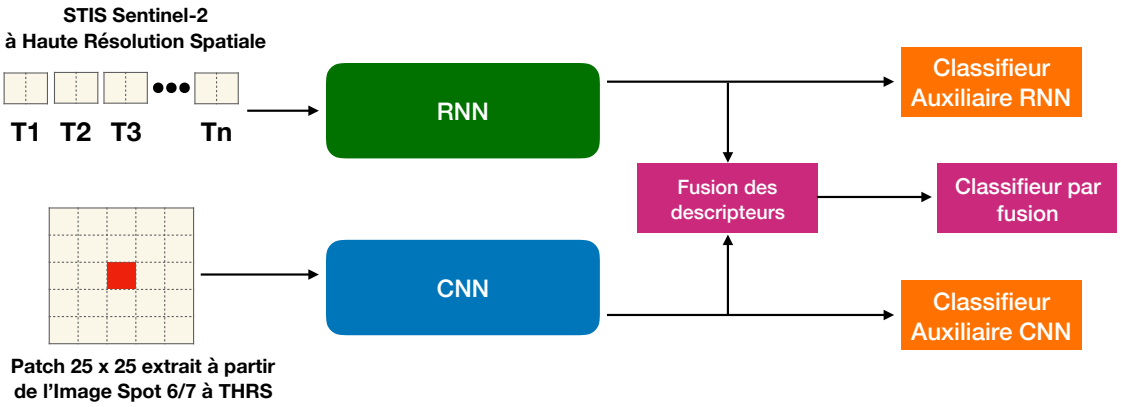

FIGURE 1 – Schéma visuel de l'approche $M^3Fusion$.

chaque date : (h_{t_1},..., h_{t_N}) où chaque h_{t_i} a la même dimension d. Si on enchaîne verticalement l'ensemble des vecteurs, nous pouvons les représenter sous forme matricielle $H \in \mathbb{R}^{N,d}$. Le mécanisme d'attention nous permet de combiner ensemble les différents vecteurs h_{t_i}, dans un vecteur rnn_{feat}, pour mieux exploiter l'information élaborée aux différents estampilles temporelles par l'unité GRU. Plus précisément, la formulation d'attention que nous avons utilisée à partir d'une séquence de vecteurs des descripteurs entraînés (h_{t_1},..., h_{t_N}) est :

$$v_a = tanh(H \cdot W_a + b_a) \quad (1)$$

$$\lambda = SoftMax(v_a \cdot u_a) \quad (2)$$

$$rnn_{feat} = \sum_{i=1}^{N} \lambda_i \cdot h_{t_i} \quad (3)$$

La matrice $W_a \in \mathbb{R}^{d,d}$ et les vecteurs $b_a, u_a \in \mathbb{R}^d$ sont des paramètres entraînés pendant l'apprentissage. Ces paramètres permettent de combiner les vecteurs de la matrice H. Le but de cette procédure est de déduire un ensemble de poids (λ_{t_1},..., λ_{t_N}) qui permettent de pondérer la contribution de chaque estampille temporelle h_{t_i} dans une combinaison linéaire. La fonction $SoftMax(\cdot)$ [9] est utilisée pour normaliser les poids λ de façon à ce que leur somme soit égale à 1. La sortie du module RNN est le vecteur rnn_{feat}, ce vecteur encode l'information relative à ts_i pour le pixel i.

3.3 Intégration de l'information à Très haute Résolution Spatiale (THRS)

L'information à Très Haute Résolution est intégrée dans $M^3Fusion$ au travers l'utilisation d'un module CNN. La littérature en vision artificielle propose de nombreuses architectures de type convolutif [6, 8]. La plupart de ces réseaux sont conçus pour traiter des images RGB (trois canaux) avec une taille supérieure à 200×200. De tels réseaux sont composés de plusieurs couches (voir des dizaines ou des centaines). Dans notre scénario, les morceaux d'image à analyser ont une taille de 25×25 et contiennent cinq canaux. Afin de proposer un module CNN adapté à notre scénario et de taille raisonnable vis-à-vis du nombre de paramètres, nous avons conçu le module CNN reporté en Figure 2.

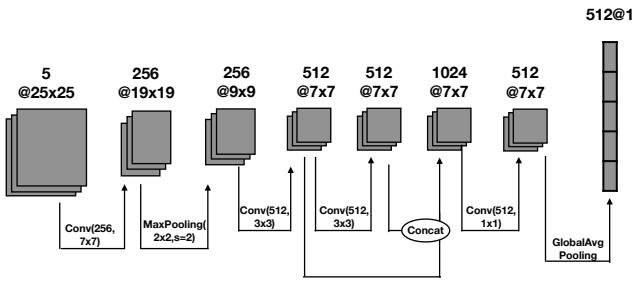

FIGURE 2 – Structure du Réseau de Neurones Convolutifs.

Notre réseau CNN applique un premier filtre 7×7 au morceau d'image à cinq canaux pour produire 256 feature maps. Ensuite un *max pooling* est utilisé pour réduire la dimension et le nombre de paramètres. Deux opérations de convolutions successives avec un noyau 3×3 extraient 512 features maps chacune, qui sont ensuite concaténées et réduites à nouveau par un filtre convolutif 1×1 (taille finale 512×7×7). Une opération de Global Average Pooling nous permet de produire un vecteur de descripteurs de taille 512 pour chaque feature map.

Chaque opération de convolution est associée à un filtre linéaire, suivi par une transformation non linéaire de type Rectifier Linear Unit [18] et d'une étape de Batch Normalization [11]. Les deux points clés de notre proposition sont : a) le nombre de filtres plus élevé à la première étape et, b) la concaténation des features maps à différentes résolutions. Le premier point est lié à la quantité d'information spectrale en entrée (cinq canaux) plus élevée par rapport aux images RGB. Pour mieux exploiter la grande richesse spectrale de ces données, nous avons augmenté le nombre de

feature maps générées à cette étape. Le deuxième point, relatif à la concaténation des features maps, vise à exploiter des informations à différentes résolutions pour extraire l'ensemble de descripteurs produits par le module CNN en suivant une philosophie similaire à [8]. La sortie de ce module est un vecteur de dimensions 512 (cnn_{feat}) qui résume le contexte spatial ($patch_i$) associé au pixel Sentinel-2 i.

3.4 Fusion des descripteurs au travers d'un processus « de bout en bout »

Un des avantages des méthodes d'apprentissage profond, comparées aux méthodes standard de machine learning, est la possibilité de pouvoir relier, dans un seul pipeline, l'extraction des descripteurs et le classifieur associé [21]. Cette caractéristique est particulièrement importante dans un processus de fusion multi-source, multi-échelle et multi-temporelle. $M^3Fusion$ exploite cet atout pour pouvoir extraire des descripteurs complémentaires à partir de deux sources d'information qui décrivent, selon des points de vue différents, le même pixel. Pour renforcer encore plus la complémentarité et, par conséquent, le pouvoir discriminant des descripteurs entraînés pour chaque flot d'information, nous avons adapté la technique proposée dans [7] à notre contexte. Dans ce travail, les auteurs ont proposé d'apprendre deux représentations complémentaires (au travers de deux réseaux convolutifs) à partir de la même image. Le pouvoir discriminant est amélioré grâce à deux classifieurs auxiliaires liés à chaque groupe de descripteurs en plus du classifieur qui utilise l'information fusionnée avec une opération de somme. Dans notre cas, nous avons deux sources d'information complémentaires par nature (séries temporelles Sentinel-2 et voisinage spatial plus précis du pixel à 10m) auxquelles deux classifieurs auxiliaires sont rattachés pour augmenter, de façon indépendante, leur habilité à reconnaître les classes d'occupation du sol. Concernant le classifieur dédié à la fusion, nous avons réalisé cette étape en concaténant les descripteurs en sortie du module CNN (cnn_{feat}) et les descripteurs en sortie du module RNN (rnn_{feat}). Le processus d'apprentissage implique l'optimisation des trois classifieurs en même temps, un premier spécifique aux descripteurs rnn_{feat}, un second spécifique aux descripteurs cnn_{feat} et le troisième spécifique aux descripteurs $[rnn_{feat}, cnn_{feat}]$.
La fonction de coût associée à notre modèle est la suivante :

$$L_{total} = \alpha_1 * L_1(rnn_{feat}, W_1, b_1) + \quad (1)$$

$$\alpha_2 * L_2(cnn_{feat}, W_2, b_2) + \quad (2)$$

$$L_{fus}([cnn_{feat}, rnn_{feat}], W_3, b_3) \quad (3)$$

où

$$L_i(feat, W_i, b_i) = L_i(Y, SoftMax(feat \cdot W_i + b_i)) \quad (4)$$

Y est la vraie valeur de la variable de classe. $L_1(rnn_{feat}, W_1, b_1)$ (resp. $L_2(cnn_{feat}, W_2, b_2)$) est la fonction de coût du premier (resp. deuxième) classifieur auxiliaire qui prend en entrée le groupe de descripteurs produit par un module spécifique (CNN ou RNN) et également les paramètres permettant d'effectuer la prédiction (W_1, b_1 ou W_2, b_2). $L_{fus}(cnn_{feat}, rnn_{feat}, W_3, b_3)$ est la fonction de coût spécifique au classifieur qui utilise la totalité des descripteurs ($[cnn_{feat}, rnn_{feat}]$). Cette dernière fonction de coût est paramétrée à travers W_3 et b_3. Chacune des fonctions de coût est modélisée à travers l'entropie croisée catégorielle (categorical cross entropy), choix typique pour des problèmes de classification supervisée multi-classe [9]. L_{total} représente la fonction de coût optimisée au travers d'un processus « bout en bout » d'apprentissage du modèle $M^3Fusion$. Une fois le réseau entraîné, la prédiction est effectuée en utilisant uniquement le classifieur qui exploite l'ensemble de tous les descripteurs paramétrés avec W_3 et b_3. Les fonctions de coût L_1 et L_2, comme souligné dans [7], constituent aussi une sorte de régularisation du modèle qui force, à l'intérieur du réseau, les descripteurs extraits par source d'information à être discriminants de façon indépendante.

4 Expérimentations

Dans cette section, nous présentons le protocole expérimental que nous avons utilisé et discutons les résultats obtenus sur le jeu de données présenté en Section 2.

4.1 Protocole Expérimental

Nous comparons les performances de l'approche d'apprentissage profond $M^3Fusion$, que nous proposons, par rapport au classifieur Random Forest (***RF***) qui est communément utilisé pour la classification supervisée dans le domaine de la télédétection [14].
Pour le modèle ***RF***, nous avons fixé le nombre d'arbres aléatoires générés à 200 sans spécifier de limite sur leur profondeur. Pour le Random Forest, nous avons utilisé l'implémentation python fournie par la bibliothèque Scikit-learn [19]. Pour pouvoir comparer les deux méthodes, nous fournissons en entrée de la méthode ***RF*** les mêmes données que celles fournies à la méthode $M^3Fusion$. Chaque exemple du jeu de données pour ce compétiteur a une taille de 3 669 qui correspond à $25 \times 25 \times 5$ ($patch_i$) plus 34×16 (ts_i).
Pour notre modèle nous avons choisi la valeur d (nombre d'unité cachée dans l'unité récurrente GRU) à 1 024. De façon empirique nous avons fixé α_1 et α_2 à 0.3. En ce qui concerne la phase d'apprentissage, nous avons utilisé la méthode Adam [13] pour apprendre les paramètres du modèle avec un taux d'apprentissage égal à $2 \cdot 10^{-}4$. Le processus d'entraînement est mené sur 400 époques. Le meilleur modèle par rapport à la valeur de la fonction de coût est utilisé en phase de test.
Nous avons implémenté $M^3Fusion$ à l'aide de la bibliothèque python Tensorflow. La phase d'apprentissage du modèle prend environ 15h et, la classification sur le jeu de

données de test dure moins d'une minute environ sur une station de travail avec un processeur Intel (R) Xeon (R) CPU E5-2667 v4@3.20Ghz avec 256 Go de RAM et GPU TITAN X.
Concernant les données, nous avons divisé le jeu en deux parties, une pour l'apprentissage et une autre pour la validation des performances des méthodes de classification supervisée. Nous avons aussi utilisé 30% des objets pour l'entraînement (soit 97 110 pixels) et les 70% restants pour le test (soit 225 638 pixels). Nous avons veillé à ce que les pixels du même objet appartiennent exclusivement à l'ensemble d'entraînement ou à l'ensemble de test [10]. Les valeurs ont été normalisées, par bande spectrale, dans l'intervalle $[0, 1]$.
En raison du déséquilibre entre les classes d'occupation du sol en termes de nombre d'échantillons, le *F-Measure* est utilisé en plus de la précision globale (*Accuracy*) pour évaluer les performances de classification [9].

4.2 Résultats Quantitatifs

La Figure 3 montre les résultats des deux méthodes de classification en termes de F-Measure par classe. Nous pouvons observer que la méthode d'apprentissage profond obtient des résultats meilleurs ou comparables à la méthode *RF* sur toutes les classes. Nous constatons que sur la classe (12) (où le *RF* est légèrement meilleur) les résultats sont similaires. Par contre, pour les autres classes, que nous pouvons considérer comme étant plus difficiles à traiter car les performances absolues sont moins bonnes, l'augmentation des performances obtenues par la méthode d'apprentissage profond est considérable. Ces classes sont : (1),(3),(4),(5),(7),(8) et (9).

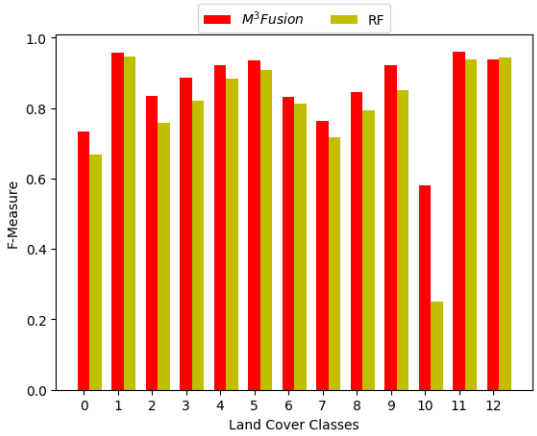

FIGURE 3 – F-Measure par classe des deux méthodes de classification.

Une classe sur laquelle l'amélioration est nettement plus sensible est la classe (10) *Cultures sous serre*. Pour une F-Measure de 0.25 fournie par *RF*, $M^3Fusion$ arrive à atteindre une F-Measure de 0.58. En effet, l'aspect et la dynamique des éléments de cette classe sont très proches de ceux du bâti, et seule l'augmentation de l'espace de représentation des données apporté par l'approche profonde permet de dériver des caractéristiques plus discriminantes [9].
En ce qui concerne les résultats de la mesure Accuracy, $M^3Fusion$ (resp. *RF*) a réalisé une classification exacte à 90.67% (resp. 87.39%). Pour une analyse plus fine, nous avons reporté en Figure 4a (resp. Figure 4b) la carte de chaleur associée à la matrice de confusion de la méthode *RF* (resp. de la méthode $M^3Fusion$). Celle-ci donne un bon aperçu des comportements des deux méthodes sur le jeu de données. Tout d'abord nous pouvons observer que la carte de chaleur du *Random Forest* est plus bruitée, en particulier autour de la diagonale. Ce bruit localise les erreurs du classifieur dans sa décision. Ce comportement est particulièrement évident sur la classe (10) où la majorité des éléments de cette classe est positionnée dans la classe (9). Pour la méthode $M^3Fusion$, nous pouvons observer dans la carte de chaleur une structure plus claire le long de la diagonale avec moins de bruits. Notre méthode a tendance à confondre les deux classes (9) et (10), mais ce comportement est moins prononcé par rapport à la méthode *Random Forest* et la plupart des éléments sont mieux classés.

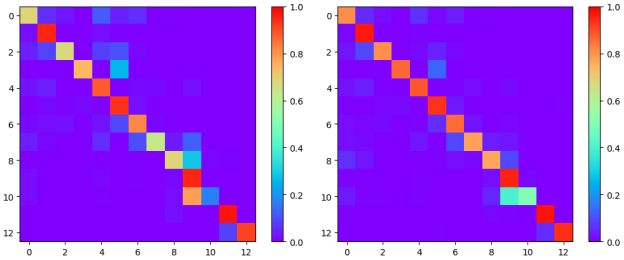

FIGURE 4 – Matrice de Confusion de la méthode *Random Forest* (a) et de la méthode $M^3Fusion$ (b)

Les résultats présentés jusqu'à présent sont liés à un seul découpage 30%/70% du jeu de données. Il est connu que selon le découpage des données, les performances des différentes méthodes peuvent varier car des exemples plus simples ou plus difficiles peuvent être insérés dans l'ensemble d'entraînement ou dans celui de test. Afin d'avoir une première compréhension de la robustesse de notre méthode par rapport à ce phénomène, nous avons produit quatre autres découpages du jeux de données en respectant toujours le même protocole. Les résultats sur les cinq découpages sont reportés dans le Tableau 2. Nous pouvons constater que les comportements des deux méthodes, par rapport aux différents découpages, sont similaires : les deux méthodes obtiennent la meilleure performance sur le découpage 2 et les moins bons résultats sur le découpage 3. Ce comportement est lié au phénomène que nous avons évoqué plus tôt. En revanche, nous pouvons constater que la méthode $M^3Fusion$ obtient toujours les meilleurs résultats sur tous les découpages avec un gain en Accuracy (resp. en F-Measure) qui varie entre 2.28 et

Essai	*RF*		$M^3Fusion$		*Gain*	
	Acc.	*F-Meas.*	*Acc.*	*F-Meas.*	*Acc.*	*F-Meas.*
1	87.39	87.11	90.67	90.67	+3.28	+3.56
2	88.47	88.05	91.52	91.39	+3.05	+3.34
3	85.21	84.62	89.25	89.15	+4.04	+4.53
4	88.33	88.05	90.61	90.7	+2.28	+2.65
5	87.29	86.88	90.09	89.96	+2.8	+3.08

TABLE 2 – Résultats de Accuracy et F-Measure des deux méthodes de classification sur cinq découpage différents du jeux de données.

4.04 (resp. 2.65 et 4.53). Nous pouvons souligner deux autres points remarquables, $M^3Fusion$ semble être plus stable que la méthode *Random Forest*. Nous observons ce comportement sur le découpage numéro 3 où les performances du classifieur propositionnel diminuent de plus de 3 points par à rapport à son meilleur résultat. Dans le cas de $M^3Fusion$, la différence entre le meilleur résultat et le plus mauvais est autour de 2 points. Enfin nous pouvons encore remarquer que, pour les résultats relatifs à la méthode d'apprentissage profond, l'écart entre la valeur de Accuracy et la valeur de F-Measure est minimum, les deux valeurs sont toujours plutôt similaires et alignées. Au contraire, pour la méthode *Random Forest* nous pouvons observer un certain écart entre les deux mesures. La mesure de Accuracy est toujours plus élevée d'environ un demi-point. En regardant les résultats de plus près, nous avons constaté que cette méthode paraît plus sensible au déséquilibre entre les classes et, semble privilégier les classes majoritaires dans sa décision.

4.3 Résultats Qualitatifs

En plus des évaluations numériques présentées dans la section précédente, nous proposons également une première évaluation qualitative de la carte produite par la méthode $M^3Fusion$. La carte obtenue par la méthode $M^3Fusion$ est également reportée en Figure 5 pour un aperçu qualitatif. La reconnaissance des classes majoritaires, à savoir les surfaces cultivées en canne à sucre sur la côte, les différents gradients d'espaces naturels (prairies, savanes et forêts) ainsi que le tissu urbain, semble bien localisée et régulière, avec une présence de bruit moins importante que sur la carte obtenue par *Random Forest* (non reportée pour brièveté).
Quelques extraits remarquables des deux cartes sont fournis pour comparaison en Figure 6 : dans la ligne en haut, un extrait de tissu urbain est affiché. La présence de bruit est particulièrement marquée sur la carte RF (au milieu), les zones situées entre les bâtiments sont souvent interprétées comme du maraîchage. Cette erreur n'apparaît pas sur la carte $M^3Fusion$(à droite). Dans la ligne du bas, nous pouvons noter que la carte RF est sensible à la présence de nuages ou ombres associées sur l'image THRS. Ces artefacts ne se retrouvent pas avec la méthode proposée : ceci est probablement du à un biais du RF en faveur de l'information provenant de la THRS, situation qui ne se vérifie pas avec l'approche proposée.

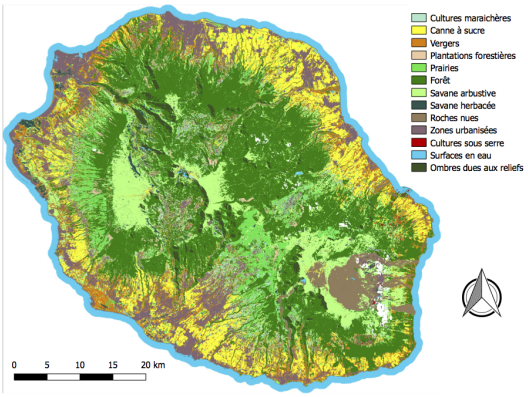

FIGURE 5 – Carte produite avec la méthode $M^3Fusion$

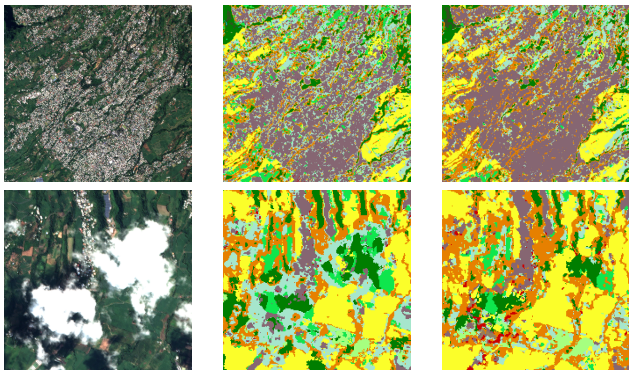

FIGURE 6 – Détails des classifications obtenues avec *Random Forest* et $M^3Fusion$: de droite à gauche, extrait de l'image SPOT6/7, classification par *RF*, classification par $M^3Fusion$.

5 Conclusions

Dans cet article, nous avons proposé une nouvelle architecture de deep learning pour la fusion de données satellitaire à haute résolution temporelle/spatiale avec des données à Très Haute Résolution Spatiale dans un but de prédiction de l'occupation du sol. Les expériences conduites sur un site d'étude réel nous ont permis de valider et d'analyser notre approche par rapport à une approche d'apprentissage automatique fréquemment utilisée dans le domaine de la télédétection. A l'avenir, nous envisageons d'étudier l'extension de notre architecture pour prendre en compte d'autres sources de données complémentaires.

6 Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre des programmes d'Investissements d'Avenir portant les références ANR-16-CONV-0004 et le projet GEOSUD

portant la référence ANR-10-EQPX-20, ainsi que de la contribution financière du compte d'affectation spéciale ≪Développement agricole et rural≫ du Ministère de l'agriculture. Ce travail a utilisé également une image acquise dans le cadre du dispositif CNES Kalideos (site de La Réunion).

Références

[1] N. A. Abade, O. Abílio de Carvalho Júnior, R. Fontes Guimarães, and S. N. de Oliveira. Comparative analysis of modis time-series classification using support vector machines and methods based upon distance and similarity measures in the brazilian cerrado-caatinga boundary. *Remote Sensing*, 7(9) :12160–12191, 2015.

[2] Y. Bengio, A. C. Courville, and P. Vincent. Representation learning : A review and new perspectives. *IEEE TPAMI*, 35(8) :1798–1828, 2013.

[3] D. Britz, M. Y. Guan, and M.-T. Luong. Efficient attention using a fixed-size memory representation. In *EMNLP*, pages 392–400, 2017.

[4] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

[5] O. Hagolle, M. Huc, D. Villa Pascual, and G. Dedieu. A Multi-Temporal and Multi-Spectral Method to Estimate Aerosol Optical Thickness over Land, for the Atmospheric Correction of FormoSat-2, LandSat, VENμS and Sentinel-2 Images. *Remote Sensing*, 7(3) :2668–2691, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] S. Hou, X. Liu, and Z. Wang. Dualnet : Learn complementary features for image recognition. In *IEEE ICCV*, pages 502–510, 2017.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

[9] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE GRSL*, 14(10) :1685–1689, 2017.

[10] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes. Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sensing*, 9(1) :95, 2017.

[11] S. Ioffe and C. Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[12] A. Karpatne, Z. Jiang, R. R. Vatsavai, S. Shekhar, and V. Kumar. Monitoring land-cover changes : A machine-learning perspective. *IEEE Geoscience and Remote Sensing Magazine*, 4 :8–21, 2016.

[13] D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[14] V. Lebourgeois, S. Dupuy, E. Vintrou, M. Ameline, S. Butler, and A. Bégué. A combined random forest and OBIA classification scheme for mapping smallholder agriculture at different nomenclature levels using multisource data (simulated sentinel-2 time series, VHRS and DEM). *Remote Sensing*, 9(3) :259, 2017.

[15] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE TGRS*, 55(2) :645–657, 2017.

[16] D. H. T. Minh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, and P. Maurel. Deep recurrent neural networks for winter vegetation quality mapping via multitemporal sar sentinel-1. *IEEE GRSL*, Preprint(-) :–, 2018.

[17] L. Mou, P. Ghamisi, and X. X. Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE TGRS*, 55(7) :3639–3655, 2017.

[18] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML10*, pages 807–814, 2010.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

[20] M. Schmitt and X. X. Zhu. Data fusion and remote sensing : An ever-growing relationship. *IEEE Geoscience and Remote Sensing Magazine*, 4(4) :6–23, 2016.

[21] L. Zhang and B. Du. Deep learning for remote sensing data : A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4 :22–40, 2016.